

Visual Servoing for the Robotenis System: a Strategy for a 3 DOF Parallel Robot to Hit a Ping-Pong Ball

Alberto Traslosheros, José María Sebastián, Luis Ángel, Flavio Roberti, Ricardo Carelli

Abstract—This article describes a new visual servo control and strategies that are used to carry out dynamic tasks by the Robotenis platform. This platform is basically a parallel robot that is equipped with an acquisition and processing system of visual information, its main feature is that it has a completely open architecture control, and planned in order to design, implement, test and compare control strategies and algorithms (visual and actuated joint controllers). Following sections describe a new visual control strategy specially designed to track and intercept objects in 3D space. The results are compared with a controller shown in previous works, where the end effector of the robot keeps a constant distance from the tracked object. In this work, the controller is specially designed in order to allow changes in the tracking reference. Changes in the tracking reference can be used to grip an object that is under movement, or as in this case, hitting a hanging Ping-Pong ball. Lyapunov stability is taken into account in the controller design.

I. INTRODUCTION

VISION systems have been more frequently incorporated in robotics applications within the last two decades. Over all, it is mainly due to visual systems advantages and the fact that computing systems are becoming more and more powerful. Visual systems have several advantages over other metrology and detection systems, and due to modern cameras and vision systems, the amount of information as well as the possibilities tend to be unlimited. On the contrary, there are visual algorithms that present extraordinary results, but with a strong disadvantage, as there can be an excessive processing time, or the fact that the environment needs to be structured in order to increase the reliability of algorithms. In the robotics areas, it is particularly interesting the use of position and orientation of objects for the robot tasks. Thus, visual information must be processed in order to extract the information that is required for the robot system [16]. Integration of visual systems and robots in dynamic tasks presents several unsolved issues that have been an object of research in important investigation centers [1, 2]. Those issues include the design of control

strategies of robotic high speed visual tracking tasks, such as in the Tokyo University ([3, 4] and more recently [5]), where fast tracking (up to 2 m/s) strategies in visual servoing have been developed.

In order to study and implement different strategies of visual control, the Computer Vision Group at the Polytechnic University of Madrid decided to design and implement the Robotenis system. The platform (shown in Fig. 1) is a parallel robot of three degrees of freedom inspired in the delta robot, with an open control structure, and equipped with a system of computer vision for image acquiring and analysis. The system has been built in order to interact with mobile objects in dynamic environments to carry out high-speed and dynamic tasks. This article presents a new visual servo control algorithm for dynamic tracking.

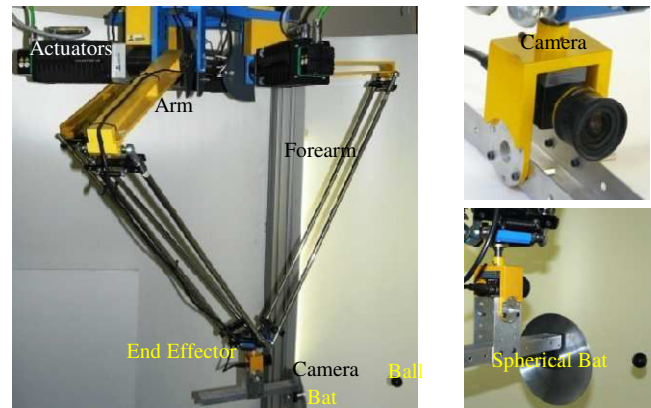


Fig. 1 Robotenis system, camera and a new spherical bat.

Roughly speaking, a parallel robot consists of two platforms linked by more than one closed kinematic chain. This kinematic structure has several advantages over serial robots: high structural stiffness, high accuracy for end effector positioning, high load operation, high speed of the end effector and high acceleration, low engine and system inertia. In contrast, the main disadvantage is the reduced range of its workspace.

The mechanical system of the Robotenis system is inspired by the DELTA robot [6], its kinematic model, the Jacobian matrix and its optimal design has been presented in a previous work [7]. The robot's kinematic structure has been optimized and its dynamics has been studied by ADAMSTM ([7, 8]). Consequently, the design method solves two problems: the determination of the dimensions of the robot and the selection of the actuators. The dynamic analysis and joint control have been presented in Angel [8] and [9]. The control system consists of two control loops

This work was partially supported by the International Cooperation Spanish Agency (AEI, A/031824/10), and the Science and Innovation Ministry (DPI2010-20863).

A. Traslosheros, J. M. Sebastián and L. Ángel are with the Universidad Politécnica de Madrid, DISAM, José Gutiérrez Abascal 2, CO 28003 Spain. Phone (+34) 913 363 061, email: atraslosheros@etsii.upm.es, jsebas@etsii.upm.es and langel@etsii.upm.es.

F. Roberti and R. Carelli are with the National University of San Juan, Av. San Martín Oeste 1109, CO 5400 Argentina. Phone (+54) 0264 4 213 303, email: froberti@inaut.unsj.edu.ar and rcarelli@inaut.unsj.edu.ar. Department of Automatics, National University of San Juan, Argentina.

that are intertwined: the first one controls the actuated joints and is executed every 0.5 ms , the second control loop is external to the first one, such is executed every $T = 8.4\text{ ms}$ (time of the visual sample), and it is based on visual information. Additionally, in the first control loop is incorporated the dynamic model (based on Lagrange multipliers) in the feedback controller. As a result the dynamic model is combined with the joint controller that is based on a PD algorithm. The second control loop is the study object of present work.

The paper is organized as follows: After this introduction, in the second section the Robotenis system is briefly described. The third section describes the proposed visual control algorithm and implementation issues. The fourth section describes some of the results concerning the measurement of the position of the ball, and practical considerations to implement the controller. The fifth section describes a spherical bat usage and the sixth section gives the conclusions.

II. SYSTEM DESCRIPTION

This section describes the Robotenis system, its function characteristics, elements and its test environment. More information can be found on [10].

A. Test Environment

The main objective of this paper is the design of a visual controller that allows playing Ping-Pong. In order to do the task, and as initial approach, a simplified environment in future works environment complication is considered. To test the algorithm, a ping pong ball is hanging on a fixed structure (Fig. (2)). The ball is able to move at a speed close to 1 m/s . The environment is simplified with a black ball on a white background in order to increase the speed of the image analysis.

B. Visual System

The Robotenis system has a single camera on the end of the robot, Fig (2). The camera location responds to two main purposes: When the object is far from the end effector, the camera has a wide view field. The second objective is that when the object and end effector are near each other, the noise has less effect over measurements, and the object's path can be accurately estimated. Those characteristics are important to our Ping-Pong application. The visual system basically consists of a light weight camera and a frame grabber. The camera SONY HCHR50 allows 240×640 pixels images at a sample time of 8.4 ms (binning mode). The progressive scan and integration time is close to 1 ms . The frame grabber is a Matrox Meteor 2-MC/4 that allows a double buffering mode acquisition, which is essential to reach the visual sample time.

C. Image Processing

Once an image is acquired, the image segmentation is done under the knowledge of the ball being on a white

background. The image features are taken from two image analysis: the first is calculated from the centroid of the ball projection, and the second is the diameter of the projected ball that is measured from the image using sub-pixel accuracy. Spatial measurements are firstly given in the camera coordinates reference, and through a second step are transformed to end effector coordinates. The camera is calibrated ([11]), and $X\ Y$ coordinates are given from the position of the ball in the image plane (centroid); the Z coordinate is calculated from the diameter of the ball. Within this work the control algorithms require knowing the position and speed of the ball, which is estimated by using a Kalman filter [12].

D. Actuators and Joint Control System

The Joint control system mainly consists of a DSPACE 1103 card where all robot algorithms are executed (implemented in ANSI C). The algorithms that are executed are the trajectory planning, kinematics and dynamics models, Dynamic and PD controller, Kalman filter (ball position and speed). The motion system is composed of three 4 kw AC brushless servomotors, Ac drivers (Unidrive) and planetary gearboxes.

E. System Special Characteristics

The Robotenis system, its design, architecture and present application determine the proposed visual controller. In the design of the visual controller, it is necessary to take into account special characteristics of the system, as it can be: noise in data that are acquired by the visual system, visual data sample time, visual data delay, actuators saturations, etc. Some of these characteristics can be a real challenge, but have to be taken into account. For example, variations in image measurements of 0.25 pixels can produce an error in the estimation of the speed of the ball near to 1 m/s (when the ball is located to 600 mm far from the camera in the z axis).

In the Robotenis system sample delay in the visual loop is estimated in 2 samples ($2T = 16.7\text{ ms}$); the visual delay is motivated by processes as well as it can be: the integration of image in the camera, the data transmission and the image processing. The saturations of the actuators, maximum Jerk, maximum acceleration, and maximum speed are taken into account in the design of a trajectory planner [13].

III. VISUAL SERVO CONTROL

In order to design the visual controller this section begins by defining the coordinated reference frames shown in Fig. 2. Σ_w , Σ_e and Σ_c are coordinate frames in the world, in the end effector of the robot and in the camera respectively. The position of the robot's end effector is defined in the world coordinate frame as ${}^w p_e$ (which is known through the robot's direct kinematics), ${}^c p_b$ is the position of the ball in the camera coordinate frame, and ${}^w p_b$ is the position of the ball in the world coordinate system. In this paper matrix, transformations and rotations are supposed to be known

through system modeling and calibrations: wR_e is the rotation matrix that relates the coordinates in the end effector to the world coordinate system, wR_c relates the camera to the world, eR_c relates the camera to the end effector, and eT_c is the homogeneous transformation that relates the camera coordinates to the end effector of the robot.

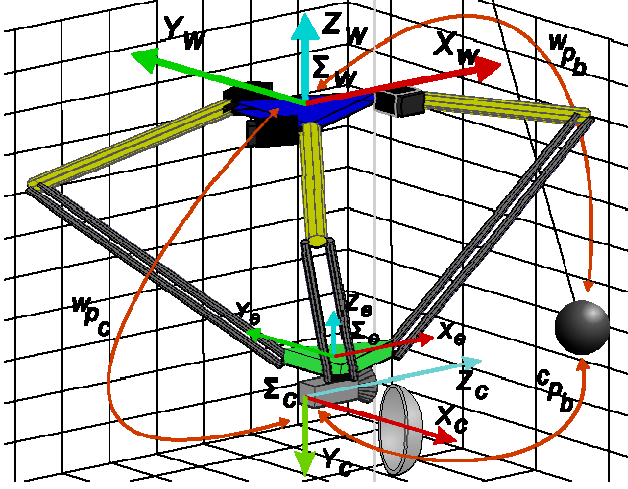


Fig. 2 Reference frames in Robotenis system.

Nowadays a wide range of controllers proposals are well established [14, 15], this article presents a visual tracking controller that is based upon a camera hand structure that is called dynamic position-based look-and-move [16]. In order to solve our problem, the PBVS scheme allows for the system to use only one single camera since the geometry of the object is known. As can be seen in Fig. 3 error can be obtained as the difference between the desired position of the ball (${}^c p_b^*(k)$) and the position that is obtained through the vision system (${}^c p_b(k)$), both in the camera reference frame and k is the sample time. In previous works desired position (${}^c p_b^*(k)$) is fixed [17]. In this paper desired position is variable and the controller stability is analyzed under this new assumption. As a result the visual controller sets the desired velocity of the end effector in order to decrease the error. Joint control signal is obtained from the visual control signal through the trajectory planner and the robot Jacobian.

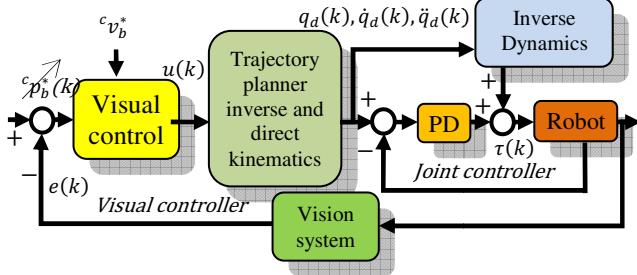


Fig. 3 Basic scheme of the controllers of the system.

A. Visual Controller Design

From Fig. 3 error can be defined as:

$$e(k) = {}^c p_b^*(k) - {}^c p_b(k) \quad (1)$$

,where ${}^c p_b^*(k)$ is the desired position and ${}^c p_b(k)$ is the measured position (measured by the visual system), both in

the camera coordinate frame. Eq. (1) can be expressed as:

$$e(k) = {}^c p_b^*(k) - {}^c R_w ({}^w p_b(k) - {}^w p_c(k)) \quad (2)$$

Differentiating eq. (2), we obtain:

$$\dot{e}(k) = {}^c v_b^*(k) - {}^c R_w ({}^w v_b(k) - {}^w v_c(k)) \quad (3)$$

From equations (2), and (3) it can be observed that ${}^c p_b^*(k)$ is not a fixed position, thus its derivative is not zero, and ${}^c v_b^*(k)$ is taken as the desired speed of approach. The desired speed of approach is used to hit the ball with a desired velocity ${}^c v_b^*(k)$ and can be approximated to:

$${}^c v_b^*(k) = ({}^c p_b^*(k) - {}^c p_b^*(k-1))/T \quad (4)$$

In order to define an error desired behavior:

$$\dot{e}(k) = -\lambda e(k) \quad (5)$$

, where $\lambda > 0$ and is a constant, and by substituting (1) and (3) in (5), we obtain:

$${}^c v_b^*(k) - {}^c R_w ({}^w v_b(k) - {}^w v_c(k)) = -\lambda ({}^c p_b^*(k) - {}^c p_b(k)) \quad (6)$$

By clearing the controller speed, we can obtain:

$${}^w v_c(k) = {}^w v_b(k) - {}^c R_w^T \left(\lambda ({}^c p_b^*(k) - {}^c p_b(k)) + {}^c v_b^*(k) \right) \quad (7)$$

The relative position and rotation between the end effector of the robot and the camera are fixed. Thus the velocity of the camera is equal to that of the end effector velocity. In contrast, the measurements from the visual system are the highly noisy, and the visual controller uses estimate data (${}^c \hat{p}_b(k)$, ${}^w \hat{v}_b(k)$). Thus, the controller output becomes:

$$u(k) = {}^w \hat{v}_c(k) = {}^w \hat{v}_b(k) - {}^c R_w^T \left(\lambda ({}^c p_b^*(k) - {}^c \hat{p}_b(k)) + {}^c v_b^*(k) \right) \quad (8)$$

B. Parameter λ

Once the controller in eq. (8) is obtained, the next step is the calculation of the parameter λ . For a sample period $k + n$, where n is small enough, it is possible to approximate as:

$${}^w p_b(k+n) \cong {}^w p_b(k) + nT {}^w v_b(k) \quad (9)$$

, and in similar form:

$${}^w p_c(k+n) \cong {}^w p_c(k) + nT {}^w v_c(k) \quad (10)$$

$${}^c p_b^*(k+n) \cong {}^c p_b^*(k) + nT {}^c v_b^*(k) \quad (11)$$

For the error equal to zero in the eq. (2) in the time $k + n$:

$$0 = {}^c p_b^*(k+n) - {}^c R_w ({}^w p_b^*(k+n) - {}^w p_c(k+n)) \quad (12)$$

Substituting (9), (10), and (11) in (12):

$$0 = {}^c p_b^*(k) + nT {}^c v_b^*(k) - {}^c R_w \left({}^w p_b(k) - {}^w p_c(k) + nT ({}^w v_b(k) - {}^w v_c(k)) \right) \quad (13)$$

From the above expression, it is possible to obtain:

$$nT {}^w v_c(k) = nT {}^w v_b(k) - {}^c R_w^T \left({}^c p_b^*(k) + nT {}^c v_b^*(k) \right) + {}^w p_b(k) - {}^w p_c(k) \quad (14)$$

Considering that in practice, ${}^c p_b(k)$, and ${}^w v_b(k)$ have to be estimated, and that ${}^w \hat{p}_b(k) - {}^w p_c(k)$ can be expressed as ${}^c R_w^T ({}^c \hat{p}_b(k))$, and substituting in (14) we have:

$${}^w \hat{v}_c(k) = {}^w \hat{v}_b(k) - {}^c R_w^T \left(\frac{1}{nT} ({}^c p_b^*(k) - {}^c \hat{p}_b(k)) + {}^c v_b^*(k) \right) \quad (15)$$

From eq. (8) and comparing to eq. (15) we can see that a good value for λ can be $\lambda = 1/(nT)$ if “ n ” is small enough.

C. Visual Controller Stability

In this section stability of the visual controller is probed by means of Lyapunov theory. It is possible to probe that under ideal conditions the error converges to zero but if conditions are not ideal then it can be proved that error is finally bounded. In order to analyze the controller stability start regarding equations (3) and (7) in order to attain the following closed loop expression:

$$\dot{e}(k) = -\lambda e(k) \quad (16)$$

A Lyapunov function is chosen as:

$$V = \frac{1}{2} e^T(k) e(k) \quad (17)$$

And from (17) and (16):

$$\dot{V} = e^T(k) \dot{e}(k) = -e^T(k) \lambda e(k) \quad (18)$$

It can be seen that eq. (18) is always negative ($\lambda > 0$) but it is known that ${}^w v_b \equiv u$ is not completely true and this implies that $\dot{e}(k)$ is not exactly equal to $\lambda e(k)$ and eq. (18) is not completely fulfilled. For this reason and it is important to consider an error ρ as:

$${}^w \hat{v}_c(k) = u(k) = {}^w v_c(k) + \rho(k) \quad (19)$$

Where $\rho(k)$ take into account the errors within estimates, and the not modeled dynamics of the system. Substituting an estimated value ${}^w \hat{v}_c(k)$ (eq. (19)) of ${}^w v_c(k)$ in eq. (3):

$$\dot{e}(k) = {}^c v_b^*(k) - {}^c R_w ({}^w v_b(k) - {}^w v_c(k) - \rho(k)) \quad (20)$$

Substituting eq. (7) in (20):

$$\dot{e}(k) = {}^c v_b^*(k) - {}^c R_w \left[{}^c R_w^T (\lambda ({}^c p_b^*(k) - {}^c \hat{p}_b(k)) + {}^c v_b^*(k)) - \rho(k) \right] \quad (21)$$

Simplifying:

$$\begin{aligned} \dot{e}(k) &= {}^c v_b^*(k) - {}^c v_b^*(k) + {}^c R_w \rho(k) \\ &\quad - \lambda ({}^c p_b^*(k) - {}^c \hat{p}_b(k)) \\ &= {}^c R_w \rho(k) - \lambda e(k) \end{aligned} \quad (22)$$

Substituting eq. (22) in our Lyapunov candidate in eq. (18):

$$\dot{V} = e^T(k) \dot{e}(k) = -e^T(k) \lambda e(k) + e^T(k) {}^c R_w \rho(k) \quad (23)$$

From above equation, $\dot{V} < 0$ is fulfilled if:

$$\|e\| > \frac{\|\rho\|}{\lambda} \quad (24)$$

Finally, if it is considered that $\rho(k) \rightarrow 0$ then the error tends to zero $e(k) \rightarrow 0$, otherwise eq. (24) is not fulfilled, and error will not decrease, but it will be bounded by:

$$\|e\| < \frac{\|\rho\|}{\lambda} \quad (25)$$

It can be observed that the error will increase, or decrease depending on the error estimates. In order to estimate ρ , it is supposed that the errors of position and velocity estimates are bigger than errors in the dynamics of the system, thus $\rho(k)$ can be obtained from equations (7), (8) and (19):

$$\begin{aligned} u(k) &= {}^w \hat{v}_b(k) - {}^c R_w^T [\lambda ({}^c p_b^*(k) - {}^c \hat{p}_b(k)) + {}^c v_b^*(k)] \\ u(k) &= \rho(k) + {}^w v_b(k) \\ &\quad - {}^c R_w^T [\lambda ({}^c p_b^*(k) - {}^c \hat{p}_b(k)) + {}^c v_b^*(k)] \end{aligned} \quad (26)$$

And $\rho(k)$ can be expressed as:

$$\rho(k) = {}^w \hat{v}_b(k) - {}^w v_b(k) + {}^c R_w^T (\lambda ({}^c \hat{p}_b(k) - {}^c p_b(k)) \quad (27)$$

, where is clear that ρ is the error in the estimations of the velocity and position.

IV. RESULTS

In order to carry out a comparison of the new controller, in this section the controller in the eq. (8) is tested, and compared with a controller shown in a previous work [17].

A. Test of the Performance of the Controller

In order to carry out the experiment, consider that (for a satisfactory motion of the system in the Cartesian space) translational axes of the end effector are decoupled, additionally in this work, rotational motions are not considered. The spherical object allows decoupling the translational movements of the end effector by using a monocular visual system. The position of the object in the plane X, Y is obtained from the centroid of the spherical object in the plane of the image, and the distance in the Z axis is taken from the diameter of the object. This makes that the performance of the controller be especially different along the Z axis (in comparison to X and Y axes). This difference is mainly due to the noise influence in this Cartesian coordinate, as it was mentioned in the section II-E. For this reason, the controller is tested individually in the coordinates: X, Y and Z . With the aim of comparing results from the new controller, a previous controller that was designed for static visual control (in previous works [17]) was compared and subjected to the same test. The previous controller is shown in eq. (28), and its design it is similarly obtained to the controller shown in this article as:

$$u(k) = {}^w \hat{v}_b(k) - {}^c R_w^T [\lambda ({}^c p_b^*(k) - {}^c \hat{p}_b(k))] \quad (28)$$

The controllers shown in eq. (8) and (28), are compared by means of an index that isolates the error between controllers by:

$$\text{Tracking error index} = \frac{\sum_k \|e_{xyz}\|}{\sum_k \|{}^w \hat{v}_b\|} \quad (29)$$

, whereas e_{xyz} is the vector of the tracking error, and ${}^w \hat{v}_b$ is the vector of the estimated velocity of the ball. With the intention of considering similar conditions, the relative

reference between the camera end, the ball is always variable within the time (in a real case this function is known since it is possible to design a trajectory to hit, or catching the object, in this case a ball).

Two kinds of test were implemented: first the ball was fixed (static case), and second the ball was constantly under movement along a predetermined axis (dynamic case). Bear in mind that the relative reference constantly shifting. Both controllers were subjected to the same experiment conditions and the same cases.

Tests were carried out by applying the index in (29) and results are shown in the table 1. The results on table 1 are the mean of 100 of different tests, such parameters of a reference between the ball and camera were constantly modified. Additionally each test is the result of $k = 5000$

visual samples. As is shown on table 1, tests were applied to each axis independently in order to analyze error per axis (due to the differences of the noise influence, especially in the Z axis -deep estimate-), whereas other axes remained relatively constant (6 tests shown in table 1). Last two tests in the table 1 are additional experiments, where the reference was variable in all axes, in the static case the ball is fixed and in the second one the ball was under movement. Additionally, in fig. 4 two experiments are shown, and are relative to the table 1. Within table 1 it is shown that the dynamic reduces the error nearly to half. In fig. 4 a), and b) the difference is notorious, the index on table 1 shows that the error is drastically reduced from 12.035 to 5.618. In fig. 4 c), and d) error is reduced from 18.659 to 11.956.

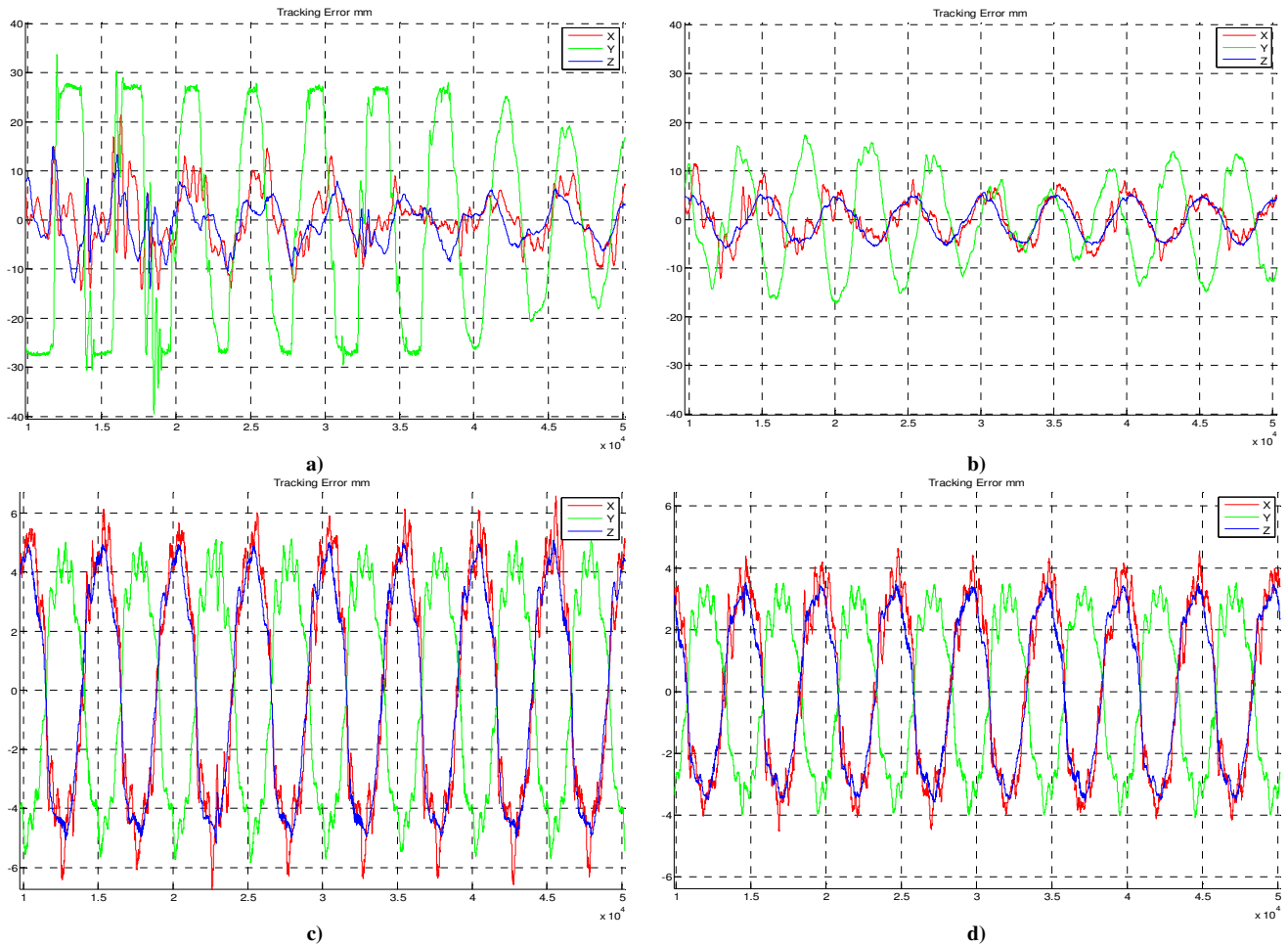


Fig. 4 shows the evolution of the tracking error in the Cartesian space. Figures a) and b) show the positioning errors (of two controllers) when the ball is fixed and the reference is shifting along the “Y” axis. Figure a) Shows errors that are produced with the controller in eq. (28), figure b) Shows errors that are produced with the controller in eq. (8). Figures c) and d) show positioning errors (of two controllers) when the ball and reference is shifting along XYZ axes simultaneously, figure c) Shows errors that are produced with the controller in eq. (28), figure d) Shows errors that are produced with the controller in eq. (8).

TABLE I

Error tracking index defined as in (29). Two kinds of ball movements are considered, the ball is fixed (static), or under movement along a predefined axis (dynamic). Additionally, the relative reference is always shifting along each axis. In the last dynamic case, the ball is moved simultaneously along the XYZ axes, and the reference being shifting along XYZ axes.

| Axis relative displacement | Tracking err. index in controller, eq. (8) | Tracking err. index in controller, eq. (28) | Type of ball movement |
|----------------------------|--|---|-----------------------|
| X | 5.795 | 12.225 | Static |
| Y | 5.618 | 12.035 | |
| Z | 5.360 | 11.742 | |
| X | 9.536 | 15.428 | Dynamic |
| Y | 9.488 | 15.358 | |
| Z | 9.679 | 15.538 | |
| XYZ | 7.869 | 13.826 | Static |
| XYZ | 11.956 | 18.659 | Dynamic |

V. STRATEGY TO HIT A PING PONG BALL

Additionally to the results shown above, in order to test the control law in (8), a ping-pong ball hitting strategy is designed and tested in the robot system. A video can be downloaded from the Robotenis webpage. Basically, the game strategy consists in hitting the ball in a position that is (estimated) known and inside of the work space of the robot. Due to the 3DOF (only traslational) of the end effector of the robot, a spherical bat was designed to hit the ball in a desired direction. Furthermore, the workspace area was divided by three planes that are perpendicular to the Z axis of the camera. Thus when the ball is “too far”, or beyond the plane 1 (respect to the Z axis of the camera, and shown in Fig. 5), the robot tracks the object only in the plane X and Y (of the coordinate system of the camera) in order to guarantee that the object is not too far from the bat of the robot. The second area is limited by the second and first plane, in this area it is possible that the ball may be out of the field of view of the camera, and here the robot decides the point where to hit the ball. Thus when the ball is out of the field of view of the camera, then the robot is guided by the estimate of the trajectory of the ball. It is important to reach out that due to the allocation of the bat, the ball is constantly out of the field of view of the robot, especially when the robot hits the ball. The third plane is the zone where the robot tracks the ball in the plane XY of the camera while it plans the trajectory, speed and the time in which the ball has to be hit (Fig. 5 is complemented in Fig. 6). Videos and an additional algorithm to allocate the bat in the desired position (XYZ) can be seen in: <http://www.disam.upm.es/vision/projects/robotenis/>

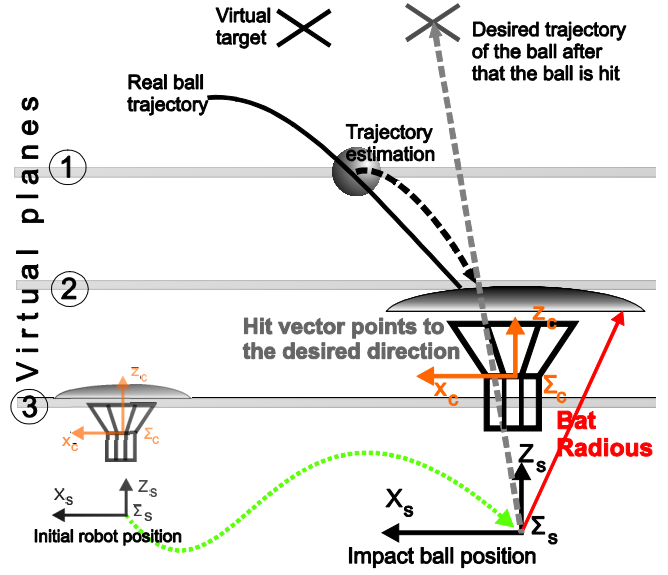


Fig. 5 Basic scheme of the controllers of the system.

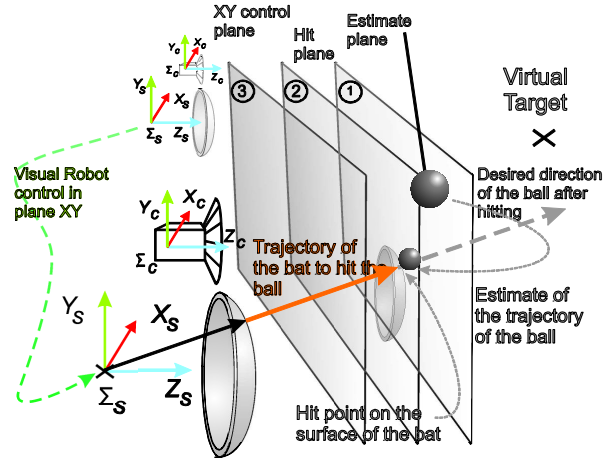


Fig. 6 Basic scheme of the controllers of the system.

CONCLUSIONS

In this paper a visual controller for dynamic tracking was presented. This controller was designed in order to carry out tasks as can be hitting, or caching objects in the 3D space. In this case the object was a black ball in a simplified scene in order to make a fast image analysis. This new visual controller application was compared with a controller designed in a previous work in order to analyze possible advantages and it was observed that error was reduced more than 40% respect in regards to the previous controller. In the web page it can be observed that the controller improves the behavior of the system and the ball when the new control law is included.

ACKNOWLEDGMENT

The authors thank the following institutions for financial support: Universidad Politécnica de Madrid in Madrid Spain; PROMEP and Universidad Autónoma de Querétaro in Queretaro, Mexico; and Universidad Nacional de San Juan in San Juan, Argentina.

REFERENCES

- [1] D. Kragic and H. Christensen, "Advances in robot vision," *Robotics and Autonomous Systems*, vol. 52, 2005, p. 1–3.
- [2] N. Oda, M. Ito, and M. Shibata, "Vision-based motion control for robotic systems," *IEEE Transactions on Electrical and Electronic Engineering*, vol. 4, 2009, p. 176–183.
- [3] M. Kaneko, M. Higashimori, R. Takenaka, A. Namiki, and M. Ishikawa, "The 100 G capturing robot- too fast to see," *IEEE/ASME transactions on mechatronics*, vol. 8, 2003, p. 37–44.
- [4] T. Senoo, A. Namiki, and M. Ishikawa, "High-speed batting using a multi-jointed manipulator," *IEEE International Conference On Robotics And Automation*, Citeseer, 2004, p. 1191–1196.
- [5] T. Senoo, Y. Yamakawa, S. Mizusawa, A. Namiki, M. Ishikawa, and M. Shimojo, "Skillful manipulation based on high-speed sensory-motor fusion," *Proceedings of the 2009 IEEE international conference on Robotics and Automation*, IEEE Press, 2009, p. 4252–4253.
- [6] R. Clavel, "Conception d'un robot parallele rapide a4 degrés de liberté", *These de Doctorat*, 1991.
- [7] L. Silva, J. M. Sebastian, R. Saltaren, R. Aracil, and J. Sanpedro, "RoboTennis: optimal design of a parallel robot with high performance," *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2005.(IROS 2005)*, Ieee, 2005, p. 2134–2139.
- [8] L. Ángel, J. M. Sebastián, R. Saltarén, R. Aracil, and R. Gutiérrez, "RoboTennis: design, dynamic modeling and preliminary control," *2005 IEEE/ASME International Conference on Advanced Intelligent Mechatronics. Proceedings*, 2005, p. 747–752.
- [9] L. Angel, J. M. Sebastian, R. Saltaren, and R. Aracil, "Robo Tennis System Part II: Dynamics and Control," *IEEE Conference On Decision And Control*, IEEE; 2005, p. 2030 - 2034.
- [10] J. M. Sebastián, "Webpage of the Robotenis System", <http://www.disam.upm.es/vision/projects/robotenis/>, 2010.
- [11] Z. Zhang, "A flexible new technique for camera calibration," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 22, 2000, p. 1330–1334.
- [12] G. Welch, Greg ; Bishop, "The Kalman Filter. Some tutorials, references, and research related to the Kalman filter," <http://www.cs.unc.edu/~welch/kalman/>, 2010.
- [13] A. Traslosheros, J. M. Sebastián, L. Ángel, F. Roberti, and R. Carelli, "Visual servoing using a parallel robot: Preliminary results," *Advanced intelligent mechatronics*, Zurich: 2007, p. 4–7.
- [14] F. Chaumette and S. Hutchinson, *Visual servo control. I. Basic approaches*, *IEEE Robotics and Automation Magazine*, vol. 13, 2006, p. 82–90.
- [15] F. Chaumette and S. Hutchinson, "Visual servo control, part II: Advanced approaches," *IEEE Robotics and Automation Magazine*, vol. 14, 2007, p. 109–118.
- [16] S. Hutchinson, G. Hager, and P. Corke, *A tutorial on visual servo control*, *IEEE Trans. Robot. Automat*, vol. 12, p. 651–670, 1996.
- [17] L. Angel, A. Traslosheros, J. M. Sebastian, L. Pari, R. Carelli, and F. Roberti, "Vision-Based Control of the RoboTennis System," *Lecture Notes In Control And Information Sciences*, vol. 370, 2008, p. 229–240.